Mult-GUP based Image Feature Matching

Liang Xu
Department of Electrical and Computer Engineering
Virginia Commonwealth University
Richmond, Virginia
Email: xul4@vcu.edu

***Abstract*—During the Ramhack this year (2016), elephant insurence proposed one of the channenges called sub-images. The goal of the challenge is to match a sub image to a big image in a image pool. The sub image is a small part of the big image. Example of the source image is Figure [1]**

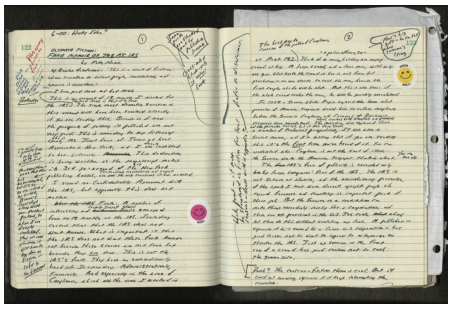

Fig. 1. Source Image

Example of the target image is Figure [2]:

Fig. 2. Target Image

During the challenge we use the eeathre matching algorithm with FLANN(Fast Approximate Nearest Neighbor Search Library) in OpenCV. OpenCV (Open Source Computer Vision Library: http://opencv.org) is an open-source BSD-licensed library that includes several hundreds of computer vision algorithms. The final challenge is to match 75 pictures. Our algorithm took around 2 hours to finish the challenge which takes too much time by using the serial computation (CPU).

So for this final project, I am thinking to continue work on the challenge using more advanced computation method, for example using GPU to parallel the computation.

CUDA and open-CV will be used, and more detailed comparison will be proposed in the future investigation.

***Index Terms*—GPU, OPENCV, Picture matching**

THIS project is going to speed up the original program. The original program run about 5 hours under 400 min Hessian distance. The average running time is 18191416089 micro seconds. Let treat this as benchmark. I will propose 4 different method to speed up the program. 1. Better Programming. 2, Use Multi Thread. 3, Single GUP. 4, Multi-GUP. In the following sections, I will discuss about each of them. The original target and source each have all the 75 pictures.

I. Running Algorithm

This program is using the OpenCV library.

Keypoints

Using the SurfFeatureDetector to extract all the keypoints. Figure [3], Figure [4]

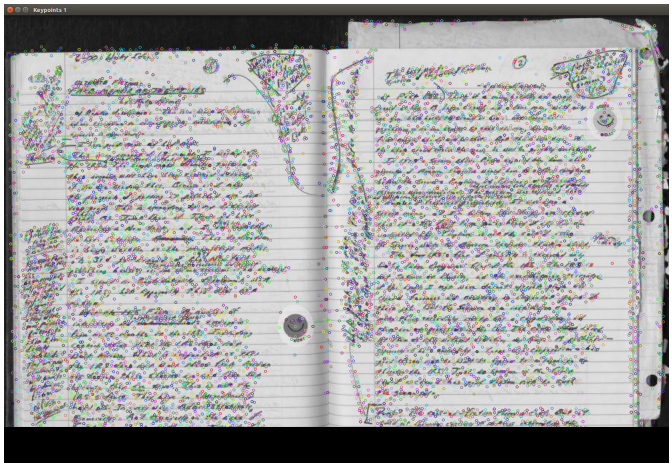

Fig. 3. Source Keypoints

Fig. 4. Target Keypoints

Hessian Value

Different Hessian value will generate different number of keypoints. Following is the three source image with different Hessian value.Figure [5], Figure [6],Figure [7]

Different Hessian value means different complexity. This project is using 400 for Hessian value.

A. descriptors and Matching method

Figure [8],Figure [9]

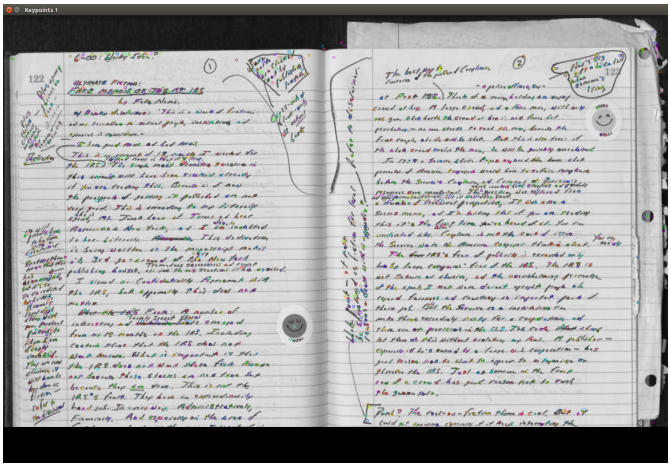

Fig. 5. Hessian = 4000

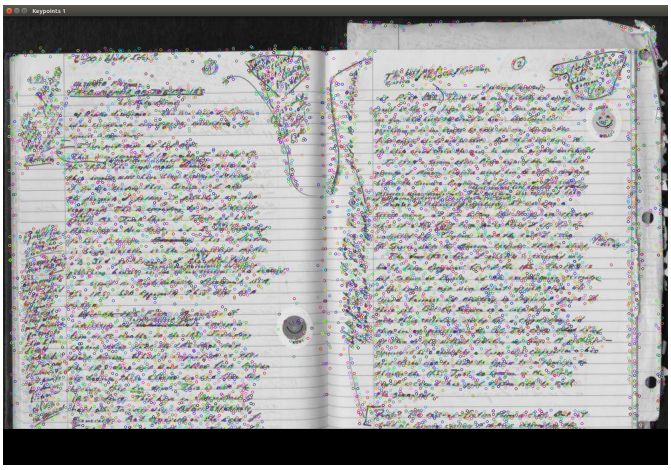

Fig. 6. Hessian = 400

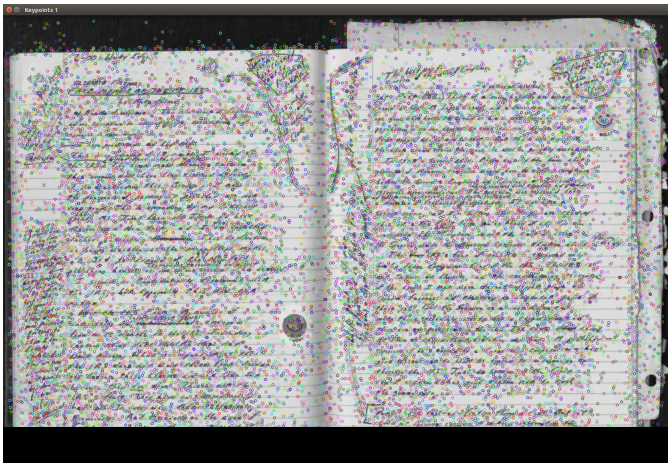

Fig. 7. Hessian = 20

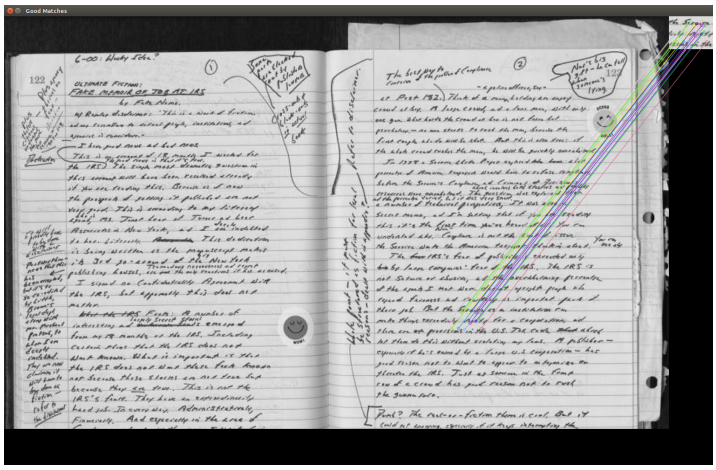

Fig. 8. Match Keypoints